AMENDMENTS TO THE SPECIFICATION:

Please amend the paragraph beginning at page 3, line 2 as follows:

The present invention, which will hereinafter be referred to as R-conversion, is an encryption system (e.g., a method, apparatus, and computer-executable process steps) for iteratively, structurally converting a binary sequence into a R-converted file of internal identifiers FID_{RC} associated with an encrypted final image G. R-conversion may be applied to any binary sequence. It is based on the concept of self-defined data. That is, an alphabet of transformation AV and a set of internal identifiers K are derived from, and are used to manipulate, the binary sequence to be converted. The ~~firmness~~ security of the method is determined at least in part by the length of all keys, and the number of conversion process iterations, algorithms and alphabets, as well as a scrambling function.

Please amend the paragraph beginning at page 3, line 17 as follows:

In one aspect, the invention comprises a method for structurally converting a binary sequence into an encrypted final image G. A first step in this method involves forming an image M of the binary sequence to be converted as a concatenation of two data elements, a tag data element T and a structural data element S. The tag data element T is comprised of information necessary to reverse the R-conversion process (or "decipher"). The structural data element S is comprised of a sequence of logical scales of position coding. A number of conversion function iterations P to be performed is then selected. The selection of P could be stochastic, but alternatively greater method ~~firmness~~ security could be achieved by selecting a larger number of iterations P. Then, the following conversion function steps are iteratively performed P times, although not all of the following steps are required to be performed upon every iteration, or in this order:

Please amend the paragraph beginning at page 10, line 13 as follows:

Referring to **FIG 3**, the ~~firmness~~ security of the encryption system is determined primarily by the lengths of keys and internal identifiers used, a number of conversion function iterations P **13**, quantization parameters m_i **36**, transformation algorithms A_i **48**, transformation alphabets AV_i **46**, and a scrambling function **11**. Each of these variables may be selected

stochastically. However, R-conversion allows for the establishment of constraining criteria (e.g., logical, mathematical, or file size) that could ~~effect~~ affect the selection of these variables. For example, if a constraining criterion is present, such as a requirement to obtain an encrypted final image G of minimum file size, the algorithms A_i **48** and quantization parameters m_i **36** (which determine alphabet AV as described below) may require re-selection based on statistical analyses of the initial binary sequence and/or computations of all possible final image G file sizes resulting from various combinations of algorithms A_i **48** and quantization parameters m_i **36**.

Please amend the paragraph beginning at page 12, line 3 as follows:

FIG. 3 represents a simplified view of the R-conversion system. In this aspect, a number of iterations P **13** has been selected, either stochastically or in accordance with a greater or lesser desired method ~~firmness~~ security. Blocks representing conversion function iterations produce converted images, such as image M'_1 **52**, which is a converted image of initial image M_0 **54**, and (optionally) extracted internal identifiers K_i **42**. The conversion function applies selected transformation algorithms A_i **48** and a self-defined transformation alphabets AV'_1 **56** to the input image, such as M_0 **54**, to obtain converted images, such as M'_1 **52**. The transformation algorithms A'_i **48** may be selected from a predefined set of algorithms L **60** which may be supplemented or changed periodically. The selection of the transformation algorithm A'_1 **48** may be stochastic, but alternatively may depend upon adherence to constraining criteria such as mathematical, logical or final encrypted image G **64** file size criteria. If a constraining criterion is present, such as a minimal encrypted final image G **64** file size, both the transformation algorithm A'_1 **48** and quantization parameter m **36** (which determines alphabet AV as described below) may require re-selection based on statistical analyses of the input image and/or computations of all possible encrypted final image G **64** file sizes resulting from various combinations of transformation algorithms A_i **48** and quantization variables m **36**. Within the structural data element **12** of each image M_n (**52**, **54**, and **62**), there are a certain number of finite binary strings, one of which may act as an internal identifier K **42** for increasing the method's ~~firmness~~ security. Internal identifier K **42** is defined by two stochastically selected parameters (not shown in **FIG 3**), a bit length parameter **22** and a shift parameter **20**, which refer to a particular binary string within structural data element S **12**. Upon each iteration, each input image M_n (**52**, **54**, and **62**) is converted to obtain a resulting image M_{n+1}. And for each iteration of the conversion except the final iteration,

the resulting image M_{n+1} becomes the input image for the next iteration. This process continues for P **13** iterations to obtain a final encrypted image G **64**. In order to restore the initial binary sequence (i.e., to decrypt G **64**), it may be necessary to accurately and in precise sequence, consecutively execute in reverse all the iterative conversion function steps. Thus, if there were P **13** conversion iterations, it may be necessary to determine P **13** internal identifiers K_n **42**, restore P **13** images M_n **62**, and find P **13** transformation algorithms A_n **48**. Without restoration of the unknown parameters of a concrete transformation iteration (K_n, M_n, A'_n, AV'_n, and length of the image), it may be impossible to execute the next reverse transformation iteration.

Please amend the paragraph beginning at page 14, line 18 as follows:

From the transformation alphabet AV_n **46**, a subset AV'_n (for example, **56**) may be selected for inversion which meets the condition of the presented image M **14** in this alphabet (for example, maximum of redundancy). Image M_n **14** may be transformed with alphabet AV'_n **56** and a stochastically selected transformation algorithm A'_n **48** to obtain a converted image M_{n+1} **52**. There are a complex set of algorithms L_m **60** with which this transformation may be realized. One of these algorithms ($A'_n \blacklozenge L_n$) can be selected in a particular manner upon each conversion iteration. The transformation algorithm A'_n **48** and alphabet AV'_n **56** may be selected stochastically, or, alternatively, based on constraining conditions such as achieving a minimum encrypted image/file length for different values of quantization parameter m **36** and number of iterations P **13**. Selection of constraining conditions (for example, ~~firmness~~ security or file size) could be accomplished with a slider in a user interface.

Please amend the paragraph beginning at page 16, line 19 as follows:

As described above, one of the first steps involved in R-conversion is a step of forming an image M_n **74** based upon the binary sequence to be converted. Next, a step **76** of selecting a number of conversion function iterations P **13** is executed. As stated previously, the selecting of the number of conversion function iterations P **13** may be stochastic, or alternatively may be made in a manner to achieve a particular encryption ~~firmness~~ security. Preferably, a determination **78** of whether internal identifiers K_i **42** will be employed and extracted from the transformed structural data element S' **70** is made early in the R-conversion process. This determination **78** is different from the determination **104** made upon *every* iteration whether to

extract an internal identifier *K* **42** during that particular iteration. The extraction of an internal identifier *K* **42** is shown in **FIG 4** as occurring in step **106**, but it may actually occur at the beginning of every iteration, upon stochastically selected iterations, or after the last iteration, depending upon conditions of information protection and routing strategy. That is, extracting an internal identifier *K* **42** depends upon the security level desired. Extracting sharply increases security, but leads to additional costs. In a preferred embodiment, internal identifiers are extracted **106** in approximately 75% of the iterations.